2014 22nd International Conference on Pattern Recognition

Forest Species Recognition using Deep Convolutional Neural Networks

Luiz G. Hafemann[1], Luiz S. Oliveira[1], Paulo Cavalin[2]

[1]*Federal University of Parana, Department of Informatics, Curitiba, PR, Brazil*
[2]*IBM Research - Rio de Janeiro, RJ, Brazil*

***Abstract*—Forest species recognition has been traditionally addressed as a texture classification problem, and explored using standard texture methods such as Local Binary Patterns (LBP), Local Phase Quantization (LPQ) and Gabor Filters. Deep learning techniques have been a recent focus of research for classification problems, with state-of-the art results for object recognition and other tasks, but are not yet widely used for texture problems. This paper investigates the usage of deep learning techniques, in particular Convolutional Neural Networks (CNN), for texture classification in two forest species datasets - one with macroscopic images and another with microscopic images. Given the higher resolution images of these problems, we present a method that is able to cope with the high-resolution texture images so as to achieve high accuracy and avoid the burden of training and defining an architecture with a large number of free parameters. On the first dataset, the proposed CNN-based method achieves 95.77% of accuracy, compared to state-of-the-art of 97.77%. On the dataset of microscopic images, it achieves 97.32%, beating the best published result of 93.2%.**

I. Introduction

Recognizing forest species is an important task in many areas. In the construction industry, it is important to validate that the correct species is being used for a given construction, to ensure that the properties of the material are known. The manufacturing process of wood products, such as tables and chairs, may require a particular type of wood. In commerce, identifying the species is important for valuing a product, and for inspection to control the illegal trade of rare species, which is an issue in many countries. Considering that these tasks generally require a human expert, the development of an automated system could lower cost and make this process faster. For this reason, several systems have been proposed in the literature for forest species recognition [1], [2], [3], [4], [5].

Forest species recognition has been generally treated as a texture classification problem, due to the property that the cross section surface of trees has different patterns on different species [1]. Texture classification techniques have been explored by several authors in recent years. Tou et al [1] investigated the usage of Gabor filters and co-occurrence matrices (GLCM). Khalid et al [2] studied the usage of Local Binary Patterns (LBP) for extracting relevant features from the images, and used a K nearest-neighbour (KNN) classifier with promising results. Paula et al. investigated the usage of color-based features and GLCM in [3], and the combination of different classifiers using GLCM, LBP, CLBP and color features in [6].

Deep learning models have been receiving increased attention in recent years. These methods are frequently setting the state-of-the-art in many domains, as reviewed by Bengio in [7]. Besides improving the accuracy on different pattern recognition problems, one of the fundamental goals of Deep Learning is to move machine learning towards the automatic discovery of multiple levels of representation. The intention is to use *raw* data (e.g. image pixels) as input to the models, and let the models learn intermediate representations - that is, let the model learn the feature detectors [7]. This is especially important, as noted by Bengio, for domains where the features are hard to formalize, such as for object recognition and speech recognition tasks. In the task of forest species classification, several alternative feature extractors have been used (as stated above), demonstrating the difficulty of finding a good representation for the problem.

Deep architectures have been widely used to achieve state-of-the-art in object recognition tasks, such as the CIFAR dataset [8] where the top published results use Convolutional Neural Networks (CNN) [9]. The tasks of object and texture classification present similarities, such as the strong correlation of pixel intensities in the 2-D space, and present some diferences, such as the ability to perform the classification using only a relatively small fragment of a texture. In spite of the similaties with object classification and we observe that deep learning techniques are not yet widely used for texture classification tasks. Kivinen and Williams [10] used Restricted Boltzmann Machines (RBMs) for texture synthesis, and Luo et al. [11] used spike-and-slab RBMs for texture synthesis and inpainting. Both consider using image pixels as input, but they do not consider training deep models for classification among several classes. Titive et al. [12] used convolutional neural networks on the Brodatz texture dataset, but considered only low resolution images, and a small number os classes.

Applying deep learning techniques to forest species recognition represent a contribution not only to this specific problem, but for texture recognition problems in general. With this approach we could learn the most appropriate feature representation for textures from data. However, one characteristic of the datasets used for this task (and also other datasets containing textures) is the high resolution of the images, in contrast with the low resolution of most object recognition databases to which deep learning has been applied. As a consequence, questions such as how to adapt the existing CNN architectures for these images and how to keep the training time acceptable are a matter of concern.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.199

The contributions of this paper are twofold. First, we present an investigation of deep learning techniques, more specifically CNN, for forest species recognition. Second, we also propose a method to deal with high-resolution texture images without changing the CNN architecture used for low-resolution images. The proposed approach has been evaluated on two forest species datasets, comparing the results with published state-of-the-art results that use traditional texture methods. The CNN-based method matched the state-of-the-art for the dataset with macroscopic images, and outperformed the best published results on the microscopic images. It is worth noting that we trained a single model that achieved this performance, while the best published results use a combination of multiple classifiers generally based on multiple feature sets.

The remainder of this paper is organized as follows. Section II describes the methodology used for this paper, including the architecture of the CNN and the proposed method to train and use CNN for forest species recognition. Section III introduces de databases considered in this paper, and presents the experimental evaluation. Finally, Section IV concludes the paper.

II. Methodology

In this section we first present the CNN architecture that has been used for this work. Afterwards, we describe the proposed method to train and use such architecture in the forest species recognition problem.

A. CNN Architecture

The deep neural network architecture used in this research was based on models that achieved high levels of accuracy on object classification tasks. In particular, it contains the repeated use of convolutional layers followed by max-pooling layers, as used by Ciresan et al. [9]. The architecture is illustrated in Figure 1.

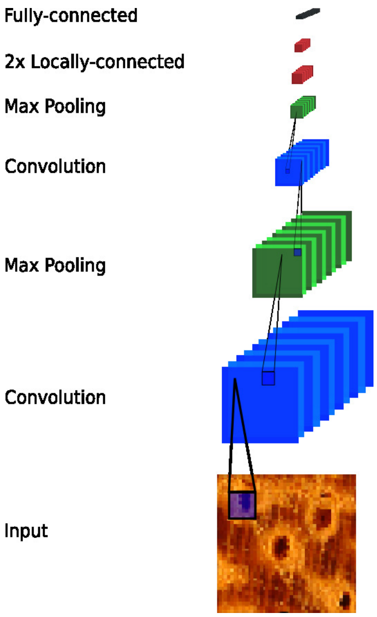

Figure 1. The Deep Convolutional Neural Network architecture.

In greater detail, this architecture consists of the following layers, with the following parameters:

1) *Input layer*: the parameters are dependent on the image resolution and the number of channels of the dataset;
2) *Two combinations of convolutional and pooling layers*: each convolutional layer has 64 filters with size 5×5 and stride set to 1, and the pooling layers consist of windows with size 3×3 and stride 2;
3) *Locally-connected layer*: 32 filters of size 3×3 and stride 1;
4) *Fully-connected output layer*: dependent on the number of classes of the problem.

For the sake of completeness, next we provide additional details on the different types of layers considered.

1) Convolutional layers: The convolutional layers have trainable filters (feature maps) that are applied across the entire image[13]. The definition of the layers include the filter size, and the stride, which is the distance between the applications of filters. The stride can be smaller than the filter size, causing the filters to be applied in overlapping windows. To select these hyperparameters, we considered the experimental results of Coates et al. [14] that analyzed the impact of filter size and stride on classification accuracy, and a filter size of 5×5 has been selected, with stride 1.

2) Pooling layers: The pooling layers implement a nonlinear downsampling function, in order to reduce dimensionality and capture small translation invariances. Scherer et al. [15] evaluate different pooling architectures, and obtained best results with max-pooling layers on object classification tasks. Based on these results, a max-pooling layer was used, with a window size of 3×3.

3) Locally-connected and Fully-connected layers: Fully-connected layers are the standard for neural networks, and connect, using unshared weights, all the neurons from one layer to the next one. Locally-connected layers only connect neurons within a small window to the next layer, similarly to convolutional layers, but without sharing weights. Combinations of both types of layers were tested, and the best results were obtained with two locally-connected layers of rectified linear units, and a final fully-connected layer with softmax activation.

B. Proposed Method For Forest Species Recognition

The proposed method aims at dealing with the high resolution of the images generally used for forest species. Adapting the existing deep neural network models for larger images can result in more complex architectures, with larger sets of parameters (more and larger layers), which can substantially increase the complexity of the model. As a consequence, the time that is necessary to fine-tune and train the parameters of the architecture can become very high, such as in the model presented by Le et al [16] in the ImageNet dataset. For this reason, we propose a method to make use of the aforementioned CNN architecture for datasets that contain images with much higher resolutions than most benchmark datasets for deep learning, e.g. 32×32 images in the CIFAR dataset and 28×28 in MNIST.

The proposed method is based on the extraction of random patches for training, and the combination of segments for recognition. More details are provided in the remainder of this section.

C. Training using random patches

To learn the parameters of the CNN described in the previous section, only small patches of the images are used for training. The main idea is to extract from the high resolution images patches with sizes that are close to those of the CIFAR and MNIST dataset. Since we are dealing with textures, the main premise is that these patches can contain enough information for training a model, provided an appropriate set of patches is extracted from each image. This strategy is similar to the one proposed by Krizhevsky et al. [17], where patches of size 224x224 are extracted from images of size 256x256. However, we take advantage of the fact that for texture datasets this patch extraction can be much more aggressive, for instance using patches that are 100 times smaller than the original texture, since for the majority of textures we can use a relatively small subset of the image to classify it.

Initial experiments were conducted to evaluate the extraction of grid patches from the images, i.e. non-overlapping patches. But we observed poor performance in this case, with the model quickly overfitting the training set. We observed, though, that good performance could be achieved with the extraction of P random patches from each image. For this reason, we adopted the following procedure. From the images in the training set, in the beginning of every training epoch we extract P randomly selected patches from each image. The dimension of the patches are the same for all images. In practice, this method is similar to the translation method used in [9], brings translation-invariance to the model and acts as regularization, preventing the model from overfitting the training set. This strategy is particularly useful for homogeneous textures, such as the majority of the forest species in the datasets considered in this paper.

D. Recognition by combining segments

For the recognition, patch results are combined for the whole image. Since the models are trained on patches of the images, we require a strategy to divide the original test images into patches, run them through the model and combine the results. The trivial solution is to use only the central patch of the image for test, but this yields poor results, since the patches are much smaller than the images. The optimal result could be achieved by extracting all possible patches from the images, but this is too computationally intensive (for the microscopic images this would generate over 300k patches for a single test image). Instead, we selected to extract the grid patches of the images, that is, the set of all non-overlapping patches, which in practice demonstrated reasonable balance between classification performance and computational cost.

Running the model, each patch outputs the probability of each possible class given the patch image. To combine the results of all the patches of a given test image, we tested the Sum rule and the Median rule, as per Kittler et al. work on combining classifiers [18]. Both methods yielded similar results. In this work we consider the Sum rule: the prediction for a given test image is the class that maximizes the sum of the probabilities on all patches of the image.

III. Experimental Evaluation

The experiments conducted to evaluate the CNN-based method considered two datasets of Brazilian forest species.

The first dataset contains macroscopic images: pictures of cross-section surfaces of the trees, obtained using a regular digital camera. This dataset consists in 41 classes, containing over 50 high-resolution (3264×2448) images for each class. The procedure used to collect the images, and details on the initial dataset (that contained 22 classes at the time) can be found in [3], and the full dataset in [6]. Examples of this dataset are presented in Figure 2.

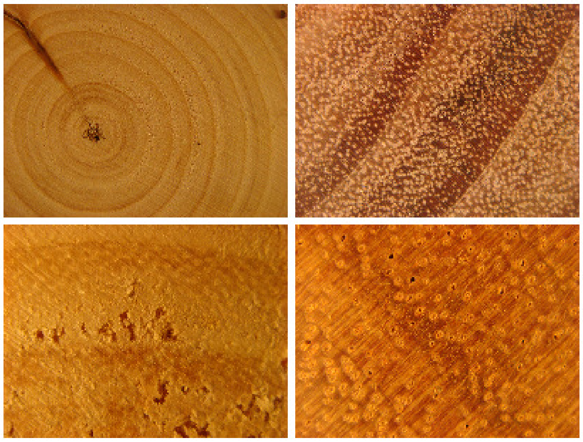

Figure 2. Sample images from the macroscopic Brazilian forest species dataset

The second dataset contains microscopic images obtained using a laboratory procedure. This dataset consists of 112 species, containing 20 images of resolution 1024×768 for each class. Details on the dataset, including the procedure used to collect the images can be found in [20]. Examples of this dataset are presented in Figure 3. It is worth noting that the colors on the images are not natural from the forest species, but a result of the laboratory procedure to produce contrast on the microscopic images. Therefore, the colors are not used for training the classifiers.

Note that the original images are significantly larger than the current datasets for object classification (3264×2448 pixels on the macroscopic dataset compared to 32×32 pixels on the CIFAR dataset). In order to reduce complexity, the image dimensions were reduced. The resize ratio was selected manually by means of a visual analysis of sample images, which were randomly selected. The visual analysis focused on reducing data dimensionality but at the same time ensuring that relevant (discriminative) features could be found in 5×5 pixel regions (the size of the filters on the first convolutional layer). Given so, the macroscopic images were resized to 256×256 pixels, and the microscopic images were resized to 640×640 pixels.

Additionally, we considered the following parameters for each dataset. Patches of size 48×48 and 64×64 were

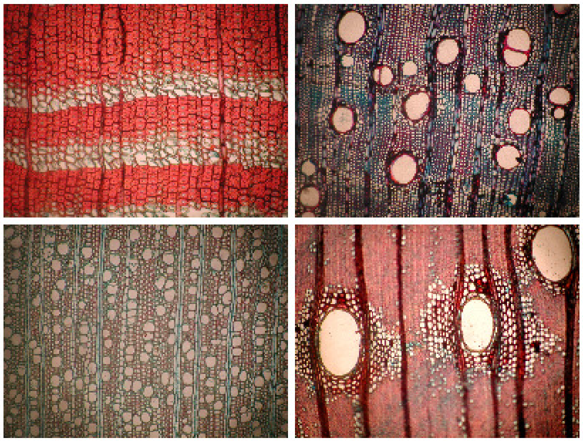

Figure 3. Sample images from the microscopic Brazilian forest species dataset

extracted from the images in the macroscopic and microscopic databases, respectively. Only a single patch was extracted from each image on each training epoch, i.e. $P = 1$. For the recognition phase, the grid division resulted in 25 patches for the first dataset, and 100 for the second one. The number of inputs and outputs of the CNNs were straightforwardly set. For the first database there are $48 \times 48 \times 3$ inputs (RGB images) and 41 outputs, while for the second there are 64×64 inputs (grayscale images) and 112 outputs.

The CNN models were trained on a Tesla C2050 GPU using the cuda-convnet library[1]. Training took about 2h for the macroscopic dataset, and about 5h for the microscopic dataset. We noticed that training took time to converge, compared to datasets with similar number of examples. This is mainly due to the fact that in each epoch, a different patch of the image is selected for training - in practice, by training for over 5000 epochs, the model sees over 5 milion different patches. Training was stopped when the error on the validation set did not improved in over 100 epochs. The model was then trained with both training and validation sets until the error on the validation set was equivalent to the error on the training set. Finally, the model was tested once in the test set. This procedure was repeated for 3 folds, and the mean and standard deviation were reported.

A. *Results*

1) Representation Learning: One of the advantages of using deep learning techniques is not requiring the design of feature extractors by a domain expert, but instead let the model learn them. We can visualize the feature detectors that model learns on the first convolutional layer, considering the weights on the learned feature maps.

Figure 4 displays the 64 feature maps learned on the first convolutional layers of both models. We can see that the model for the Macroscopic dataset learned filters for horizontal and vertical edges, and also filters that are more specific to the forest-species dataset, such as detectors for small holes in the wood. For the Microscopic dataset, there are features that resemble Gabor filters (edge detectors).

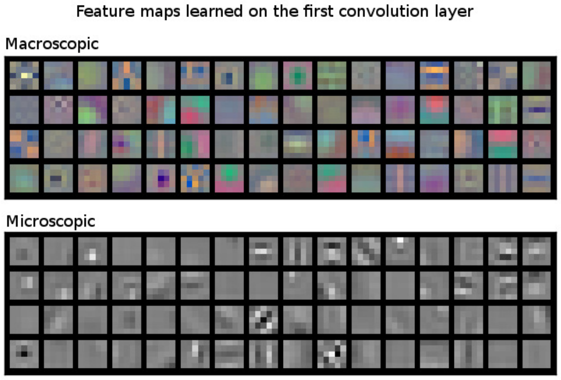

Figure 4. Feature maps (filters) learned by the first convolutional layers.

Figure 5 illustrates the models' predictions on random test examples. We can see that the models usually assign a high score when classifying the correct class, and a lower score when they misclassify an image. Only occasionally the models assign high probabilities to the incorrect classes.

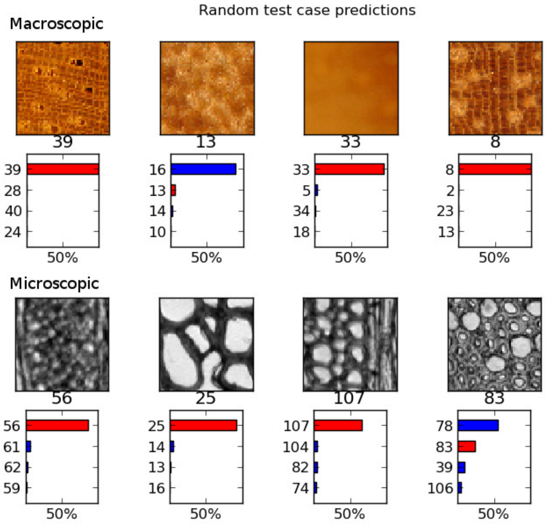

Figure 5. Random test-case predictions on the image patches. The size of the bars indicate the probability associated by the model to a given label. The red bars indicate the true labels.

2) Recognition: The recognition rates[2] of the proposed method on the Macroscopic images dataset are shown on Table I, compared to the best published in the literature, both using a single classifier or a combination of them. With recognition rates of 95.77%, the CNN-based model presents accuracy superior to other individual classifiers, except for the classifier trained with CLBP features. The best published method (that achieves an accuracy of 97.77%) combines 6 different classifiers (trained with CLBP (x2), LBP, Gabor-filters, Fractals and Color-based features).

The results on the Microscopic images dataset are shown on Table II. In this dataset, the proposed CNN-based method

[1] http://code.google.com/p/cuda-convnet/

[2] the total of correctly recognized samples divided by the total number of samples

Table I. CLASSIFICATION ON THE MACROSCOPIC IMAGES DATASET

Features and Algorithm	Accuracy
LBP (SVM) [6]	85.84%
Color-based features (SVM) [6]	87.53%
Gabor filters (SVM) [6]	87.66%
CLBP (SVM) [6]	96.22%
Multiple classifiers (SVM) [6]	**97.77%**
Proposed method (CNN)	95.77% (+- 0.27%)

outperformed the existing best classifiers, including the method that combine multiple classifiers. Our method achieved 97.32% of recognition rates, while the second best approach, combining LPQ and GLCM features and using SVM classifiers, achieved 93.2%.

Table II. CLASSIFICATION ON THE MICROSCOPIC IMAGES DATASET

Features and Algorithm	Accuracy
LBP (SVM) [20]	86.00%
GLCM (SVM) [5]	80.7%
LBP (SVM) [5]	88.5%
LPQ (SVM) [5]	91.5%
LPQ + GLCM (SVM) [5]	93.2%
Proposed method (CNN)	**97.32%** (+- 0.21%)

IV. CONCLUSIONS

In this paper we investigated the use of deep learning techniques applied to two forest species classification tasks. We presented a method to apply convolution neural networks to perform recognition on high-resolution texture images, using the same CNN architecture previously applied on object recognition tasks involving images with lower resolution. The experimental results demonstrate that the proposed deep learning-based method can achieve state-of-the-art performance on forest species datasets, achieving 95.77% of accuracy on macroscopic images, compared to state-of-the-art of 97.77%, and 97.32% of accuracy on microscopic images, surpassing the best published result of 93.2%. We also identify that the models are capable of learning useful feature detectors, capable of detecting edges, color-based features and holes in the woods.

ACKNOWLEDGEMENT

This research has been supported by the CNPq, grants #301653/2011-9 and #471050/2013-0.

REFERENCES

[1] J. Y. Tou, Y. Tay, and P. Y. Lau, “A comparative study for texture classification techniques on wood species recognition problem,” in *Fifth Int. Conf. on Natural Computation, 2009.*, 2009, pp. 8–12.

[2] M. Nasirzadeh, A. Khazael, and M. bin Khalid, “Woods recognition system based on local binary pattern,” in *2010 Second International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, 2010, pp. 308–313.

[3] P. Filho, L. Oliveira, A. Britto, and R. Sabourin, “Forest species recognition using color-based features,” in *2010 20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 4178–4181.

[4] P. Filho, “Reconhecimento de especies florestais atraves de imagens macroscopicas,” Ph.D. dissertation, Universidade Federal do Parana, 2012.

[5] P. R. Cavalin, M. N. Kapp, J. Martins, and L. S. Oliveira, “A multiple feature vector framework for forest species recognition,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2013, pp. 16–20.

[6] P. L. P. Filho, L. S. Oliveira, S. Nisgoski, and A. S. Britto, “Forest species recognition using macroscopic images,” *Machine Vision and Applications*, vol. 25, no. 4, pp. 1019–1031, 2014.

[7] Y. Bengio and A. Courville, “Deep learning of representations,” in *Handbook on Neural Information Processing*. Springer, 2013, pp. 1–28.

[8] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Master’s thesis, Department of Computer Science, University of Toronto, 2009.

[9] D. Ciresan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3642–3649.

[10] J. J. Kivinen and C. Williams, “Multiple texture boltzmann machines,” in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 638–646.

[11] H. Luo, P. L. Carrier, A. Courville, and Y. Bengio, “Texture modeling with convolutional spike-and-slab RBMs and deep extensions,” *arXiv:1211.5687*, Nov. 2012.

[12] F. H. C. Tivive and A. Bouzerdoum, “Texture classification using convolutional neural networks,” in *TENCON 2006. 2006 IEEE Region 10 Conference*. IEEE, 2006, pp. 1–4.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[14] A. Coates, H. Lee, and A. Ng, “An analysis of single-layer networks in unsupervised feature learning,” *Engineering*, pp. 1–9, 2010.

[15] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Artificial Neural Networks–ICANN 2010*. Springer, 2010, pp. 92–101.

[16] Q. V. Le, “Building high-level features using large scale unsupervised learning,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8595–8598.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks.” in *NIPS*, vol. 1, 2012, p. 4.

[18] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[19] J. Martins, L. S. Oliveira, S. Nisgoski, and R. Sabourin, “A database for automatic classification of forest species,” *Machine Vision and Applications*, vol. 24, no. 3, pp. 567–578, Apr. 2013.